OMNI-DIRECTIONAL FACE DETECTION BASED ON REAL ADABOOST

Chang HUANG[1], Bo WU[1], Haizhou AI[1]and Shihong LAO[2]
[1] Computer Science and Technology Department, Tsinghua University, Beijing, 100084, China
[2] Sensing Technology Laboratory, Omron Corporation
E-mail: ahz@mail.tsinghua.edu.cn

ABSTRACT

In this paper, we propose an omni-directional face detection method based on the confidence-rated AdaBoost algorithm, called real AdaBoost, proposed by Schapire and Singer [1]. To use real AdaBoost, we configure the confidence-rated look-up-table (LUT) weak classifiers based on Haar-type features. A nesting-structured framework is developed to combine a series of boosted classifiers into an efficient object detector. For omni-directional face detection our method has achieved a rather high performance and the processing speed can reach 217ms per 320×240 image. Experiment results on the CMU+MIT frontal and the CMU profile face test sets are reported to show its effectiveness.

1. INTRODUCTION

Recently Viola and Jones [2] have proposed a boosting method for object detection and achieved very good performance on the frontal face detection problem comparable to the best of previous systems [3]. This prompted the development of systems for more general version of face detection problems, such as rotation invariant face detection [4] and multi-view face detection (MVFD) [5]. The ultimate goal for face detection is a real-time omni-directional face detection system that can detect faces with 360° rotation-in-plane (RIP) and ±90° rotation-out-of-plane (ROP) pose changes. Although the omni-directional FD problem can be solved by rotating the image and applying some MVFD method repeatedly, the process will be time consuming and the number of false alarms will increase dramatically. Omni-directional FD is much more complicated and difficult than the previous version of face detection.

Existing work on MVFD mainly includes Schneiderman et al.'s [6] method based on Bayesian decision rule and Stan Li et al.'s [5] real-time pyramid-structured MVFD system. In this paper, we propose a novel method for omni-directional FD based on Schapire and Singer's improved boosting algorithm [1] that deals with real-valued confidence-rated weak classifiers. It is called *real AdaBoost* in order to distinguish it from what we call *discrete AdaBoost*, i.e. the original AdaBoost algorithm [7] that deals with Boolean weak classifiers. The main contributions of this paper are: 1) A confidence-rated look-up-table (LUT) type weak classifier is proposed and real AdaBoost is used to learn boosted classifiers. 2) A novel nesting-structured detector is proposed and a corresponding pose estimation method for improving overall performance is developed.

The rest of the paper is organized as follows: Section 2 introduces the real AdaBoost algorithm; Section 3 the Haar feature based LUT weak classifiers; Section 4 the nesting-structured cascade; Section 5 the view-based detectors; Section 6 the pose estimation method; Section 7 the experiment results and finally Section 8 the conclusions.

2. REAL ADABOOST

Boosting algorithms can improve the performance of a weak learner L by iteratively calling it to find a small number of weak classifiers h and then combining them into a strong one H. Real AdaBoost algorithm is a variation of AdaBoost. It deals with the confidence-rated weak classifiers that are map from a sample space X to a real-valued space R instead of Boolean prediction. It has the following form [1]:

- Given a dataset $S=\{(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_m,y_m)\}$, where $\mathbf{x}_i \in X$, and $y_i \in \{-1,+1\}$, the weak classifier pool F and the number of weak classifiers to be selected T.
- Initialize the sample distribution $D_1(i) = 1/m$.
- For $t = 1,\ldots,T$
  1. For each weak classifier h in F do:
     a. Partition X into several disjoint blocks $X_1,\ldots,X_n$.
     b. Under the distribution D_t calculate
     $$W_l^j = P(\mathbf{x}_i \in X_j, y_i = l) = \sum_{i:\mathbf{x}_i \in X_j \wedge y_i = l} D_t(i) \qquad (1)$$
     where $l=\pm1$.
     c. Set the output of h on each X_j as
     $$\forall \mathbf{x} \in X_j,\ h(\mathbf{x}) = \frac{1}{2}\ln\left(\frac{W_{+1}^j + \varepsilon}{W_{-1}^j + \varepsilon}\right) \qquad (2)$$
     where ε is a small positive constant.
     d. Calculate the normalization factor
     $$Z = 2\sum_j \sqrt{W_{+1}^j W_{-1}^j} \qquad (3)$$
  2. Select the h_t minimizing Z, i.e.

$$Z_t = \min_{h \in F} Z$$
$$h_t = \arg\min_{h \in F} Z \qquad (4)$$

3. Update the sample distribution

$$D_{t+1}(i) = D_t(i)\exp\left[-y_i h_t(\mathbf{x}_i)\right] \qquad (5)$$

and normalize D_{t+1} to a p.d.f.

- The final strong classifier H is

$$H(\mathbf{x}) = \operatorname{sign}\left[\sum_{t=1}^{T} h_t(\mathbf{x}) - b\right] \qquad (6)$$

where b is a threshold whose default is zero. The confidence of H is defined as

$$Conf_H(\mathbf{x}) = \left|\sum_{t=1}^{T} h_t(\mathbf{x}) - b\right| \qquad (7)$$

It can be seen that Eq.1 and Eq.2 define the output of the weak classifier, so all that is left to the weak learner is to partition the domain X.

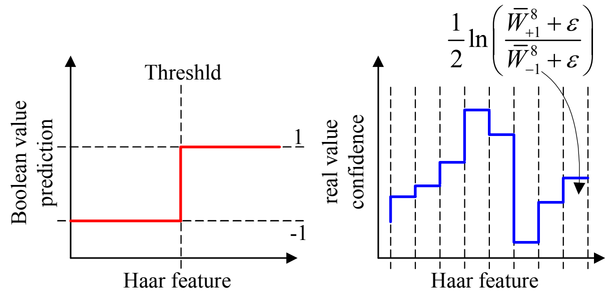

Figure 1. Threshold vs. LUT

3. HAAR FEATURE BASED LUT WEAK CLASSIFIER

Haar features are simple rectangle features used by Viola et al. [2]. For each Haar feature, one weak classifier is configured. In [2], threshold-type weak classifiers are used which output Boolean values. The main disadvantage of the threshold model is that it is too simple to fit complex distributions. Furthermore, in order to use real AdaBoost, we need a weak learner which can give a partition of X. Therefore we propose a real-valued LUT weak classifier. Assuming the Haar feature f_{Haar} has been normalized to [0,1], the range is divided into n sub-ranges:

$$bin_j = [(j\text{-}1)/n, j/n), j=1,\ldots,n \qquad (8)$$

A partition on the range corresponds to a partition on X. Thus, the weak classifier can be defined as

$$\text{If } f_{Haar}(\mathbf{x}) \in bin_j \text{ then } h(\mathbf{x}) = \frac{1}{2}\ln\left(\frac{\overline{W}_{+1}^{j}+\varepsilon}{\overline{W}_{-1}^{j}+\varepsilon}\right), \qquad (9)$$

where $\overline{W}_l^{j} = P\left(f_{Haar}(\mathbf{x}) \in bin_j,\ y = l\right),\ l = \pm 1,\ j = 1,\cdots,n$.

So it is a step function or LUT, see Figure 1.

4. NESTING-STRUCTUREED CASCADE

The cascade-structured classifier of Viola et al. [2] has been proved very efficient for object detection problems. Nevertheless there is still room for improvement. We noticed that although each layer is a boosted classifier that consists of closely coupled features, all layers in a cascade are rather loosely correlated. More specifically, each layer treats the information from its predecessor only as a binary function that tells whether or not the current sub-window is a face. It is a great waste if the confidence in Eq.7 is discarded since this value can be used to classify all preceding training samples very efficiently. Therefore, we propose a novel nesting-structured cascade, in which each layer is treated not only as an independent node of the whole cascade but also as a component of its successor, see Figure 2.

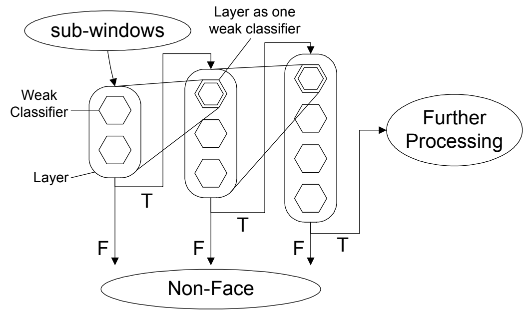

Figure 2. Nesting-structured cascade

During training, after each layer is learned, first the layer bootstraps the sample set, and then it is taken as the first weak classifier of its successor in a new boosting round. In this way, the classification ability of each layer is inherited. The training algorithm of the nesting-structured cascade is as follows:

- Given the maximum false positive rate per layer f, the minimum detection rate per layer d, the overall false positive rate F_{target}, the positive training set P and the negative training set N.
- Denote the i-th layer with L_i, and the nesting-structured detector with ND
- Initialize: $F_{all} = 1.0$, $D_{all} = 1.0$, $ND = 0$, $i = 0$
- While $F_{all} > F_{target}$
  1. $i = i + 1$
  2. If $i > 1$
     i) take $L_{i\text{-}1}$ as the first weak classifier of L_i, and update the sample distribution.
     ii) Use real AdaBoost algorithm to learn the remaining weak classifiers of L_i, until L_i satisfies the condition f and d.

     else

     Use real AdaBoost to learn L_i directly, until L_i satisfies the condition f and d.
  3. Evaluate the actual false positive rate F_i and detection rate D_i of L_i

4. $F_{all} = F_{all} \times F_i$, $D_{all} = D_{all} \times D_i$
5. $ND = ND + L_i$
6. Resample the negative training set N with ND

- ND is the final nesting-structured detector with false positive rate F_{all} and detection rate D_{all}

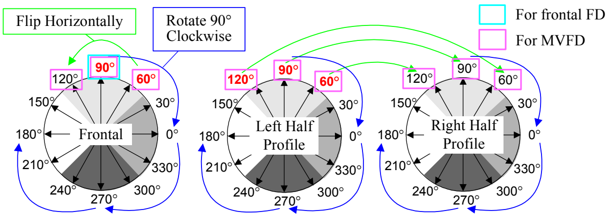

Figure 3. Flip and rotate detectors. (The red ones are the original detectors. The situation of full profile is the same as half profile.)

5. VIEW-BASED DETECTOR

According to the left-right ROP angle, multi-view faces are divided into five categories, left full profile, left half profile, frontal, right half profile and right full profile, covering [-90°, -50°], [-50°, -20°], [-20°, +20°], [+20°, +50°], [+50°, +90°] respectively. According to the RIP angle, faces are divided into twelve categories, each covering 30°. So there are in total 5 by 12 view categories corresponding to 60 detectors. Since the Haar features can be easily flipped and rotated by 90°, only eight detectors have to be trained from which all the others are generated, see Figure 3.

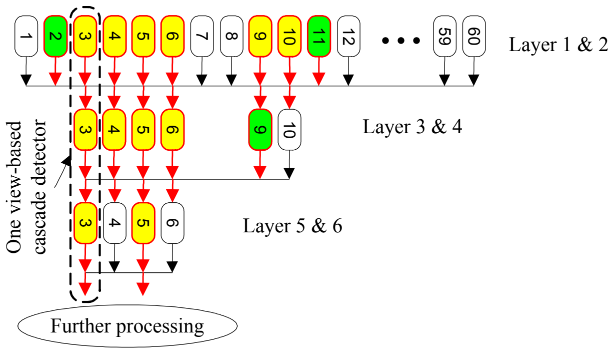

Figure 4. Pose estimation. (The block with red frame stands for the detector with positive output; the yellow block stands for the detector surviving after peeling; the green block stands for the detector peeled off.)

6. POSE ESTIMATION

When detecting face, the input image is scanned by all view-based detectors and the outputs are merged. The scanning procedure is an exhaustive search, which is very time-consuming. To improve performance in speed, we propose a pose estimation method based on the above mentioned nesting structure. The procedure of the pose estimation is like peeling an onion. The detail is as follows: According to Eq.7 each layer of every detector can output a real-valued confidence. If at the *j*th layer there are more than four detectors with positive predictions, then one fourth of them with lowest confidences will be discarded; otherwise, that is there are only four or less detectors with positive predictions, all of them will survive. The peeling operation is done several times at different layers. In our experiment, we select the 2nd, 4th and 6th layers as the peeling position, see Figure 4. So after the 6th layer, there are at most four out of all the sixty detectors can survive. It can be seen our pose estimation method is not a separate procedure from the face detection module, and therefore it will not introduce extra computation.

(a) frontal (b) left half profile (c) left full profile

Figure 5. Standard samples.

Table 1. Comparison for frontal FD on CMU+MIT frontal face test set (130 images, 507 faces, "FA" stands for false alarm and "DR" stands for detect rate)

Method	FA/DR		FA/DR	
Ours	10	90.1%	57	94.5%
Viola-Jones	10	78.3%	95	90.8%
Stan Li	10	83.6%	31	90.2%
Rowley	10	83.2%	95	89.2%

Table 2. Comparison for MVFD on CMU profile test set (208 images, 441 faces, 347 of them are non-frontal)

Method		FA/DR		FA/DR	
Ours	With PE	8	79.4%	41	85.9%
	Without PE	34	84.1%	89	86.2%
Schneiderman		12	75.2%	91	85.5%

Table 3. Results of omni-directional FD on rotated CMU profile test set (624 images, 1323 faces)

Method	FA/DR		FA/DR	
With PE	226	82.0%	342	85.8%
Without PE	423	85.0%	541	87.7%

7. EXPERIMENTS

We have collected approximately 44,000 faces including all ethnicities, a variety of ages and both genders, of which the samples are partitioned into five ROP view-based sub-categories as described in Section 5. Each category also covers ±15° RIP and ±30° up-down ROP changes. All the samples are normalized to the standard 24×24-pixel patch, see Figure 5. In our face dataset there are about 20,000 frontal faces. With these samples, we have trained a nesting-structured frontal face detector that has 16 layers and 756 Haar features. Compared to the Viola et al.'s [2] detector consisting of 4,297 features and Stan Li et al's [5] detector consisting of 2,546 features, our method is much more efficient. We have tested our system on the CMU+MIT frontal and the CMU profile face test sets [6] for frontal FD and MVFD respectively. For the former only the detector of the upright and frontal view is used and for the latter, the detectors with 60°, 90° and 120° RIP angles are used, see Figure 3. Table 1 and Table 2 show the comparison between the previous methods and ours. For omni-directional FD we rotate the CMU profile set with ±30° to generate a test set with 624 images and 1323 faces. This expanded test set covers all pose variations corresponding to the eight original detectors so that it is equivalent for the ensemble 60-detector system to test on any derivation of CMU profile set with arbitrary rotation angle within 360°. Table 3 lists the results. Figure 6 shows some detection outputs on images. When all 60 view-based detectors are active, our system can run at 217 ms per 320×240 size image, on a Pentium-IV 2.4GHz. The speedup factor of the PE is about 2.0.

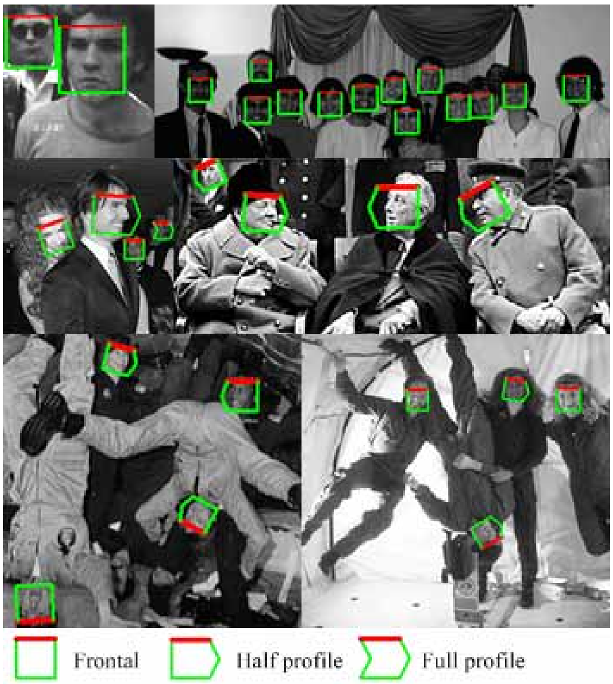

Figure 6. Some results of FD.

8. CONCLUSIONS

In this paper, we presented a method for omni-directional FD. The view-based detectors have been trained with an extensive dataset that includes faces collected from real-life photos, which implies our view based detectors are able to work at more general situations. One thing we want to mention is that in review of the latest ICCV'03 proceedings, we found that Xiao et al.'s [8] "boosting chain" originated from the same concept of inheriting previous training results, but resulted in different techniques. Theirs is a complete chain structure for discrete AdaBoost framework while ours is a nested structure for real AdaBoost. We argue that a nested structure for real AdaBoost is more powerful, and comparatively much more efficient. We have tested our method on two public test sets, the CMU+MIT frontal and the CMU profile face test sets. The latter is particularly difficult. Both of our frontal FD and MVFD systems have achieved a better performance than the previous methods. Our framework can be also applied to other object detection problems.

9. AKNOWLEDGEMENTS

This work is supported mainly by a grant from OMRON Corporation. It is also supported in part by National Science Foundation of China under grant No.60332010.

10. REFERENCES

[1] R. E. Schapire and Y. Singer, "Improved Boosting Algorithms Using Confidence-rated Predictions", Machine Learning, 37, 1999, 297-336.

[2] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA, 2001.

[3] H. A. Rowley, "Neural Network-based Human Face Detection", Ph.D. thesis, Carnegie Mellon University, May 1999.

[4] Shihong Lao, et al., "A fast 360-degree rotation invariant face detection system", in ICCV2003 Demo program.

[5] S. Z. Li et al., "Statistical Learning of Multi-View Face Detection". In Proceedings of the 7th European Conference on Computer Vision. Copenhagen, Denmark. May, 2002.

[6] H. Schneiderman and T. Kanade, "A Statistical Method for 3D Object Detection Applied to Faces and Cars". In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA, 2000.

[7] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm". In Proc. of the 13-th Conf. on Machine Learning, Morgan Kaufmann, 1996, 148-156.

[8] Rong Xiao, Long Zhu and Hongjiang Zhang, "Boosting Chain Learning for Object Detection", ICCV 2003, 709–715